ISSN 1561-5359. Штучний інтелект, 2019, № 3-4

УДК 004.93

О.С. Коваль
Інститут кібернетики імені В.М. Глушкова Національної академії наук України, Україна
пр. Академіка Глушкова, 40, м. Київ, 03187
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Україна
пр. Перемоги, 37, корпус 18, м. Київ, 03056

КОНЦЕПЦІЯ СИСТЕМИ КОМП’ЮТЕРНОГО ВІРТУАЛЬНОГО МОДЕЛЮВАННЯ МОБІЛЬНИХ РОБОТОТЕХНІЧНИХ ЗАСОБІВ НЕЙТРАЛІЗАЦІЇ ТЕХНО-ЕКОЛОГІЧНИХ ПОДІЙ ТА ВИРІШЕННЯ ЗАДАЧ ПРОФЕСІЙНОГО НАВЧАННЯ

O.S. Koval
V.M. Hlushkov Institute of Cybernetics of the NAS of Ukraine, Ukraine
40, Academician Hlushkov Ave., Kyiv, 03187
National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Ukraine
37, Peremohy Ave., korpus 18, Kyiv, 03056

CONCEPT OF SYSTEM OF COMPUTER VIRTUAL SIMULATION OF MOBILE ROBOTOTECHNICAL MEANS OF TECHNICAL AND ENVIRONMENTAL EVENTS NEUTRALIZATION AND SOLVING PROBLEMS OF PROFESSIONAL TRAINING

На сьогодні проблема віртуального проєктування, моделювання і тестування мобільних роботів до їх безпосередньої реалізації «в металі» є досить актуальною. У статті проводиться огляд технологій, що використовуються для цього, та розглядається підхід до створення спеціалізованих програмних комплексів, спрямованих на вирішення існуючих у наявних технологіях проблем. Запропоновано концепцію програми для задач моделювання структури і поведінки мобільних роботів з елементами штучного інтелекту, призначених для супроводу пошуково-рятувальних операцій у комбінації середовищ (повітряне, наземне, підводне) та для професійного навчання відповідних спеціалістів.

Ключові слова: моделювання мобільних роботів, проєктування, концепція, 3D-візуалізація, навчання, штучний інтелект

Today, the problem of virtual design, modeling and testing of mobile robots prior to their direct implementation «in the metal» is quite urgent. The article reviews the technologies used for this purpose and discusses the approach to creating specialized software complexes aimed at solving existing problems in existing technologies. The concept of the program is proposed for modeling problems of structure and behavior of mobile robots with elements of artificial intelligence, intended to support search and rescue operations in a combination of environments (air, ground, underwater) and for the professional training of relevant specialists.

Keywords: modeling of mobile robots, designing, concept, 3D-visualization, education, artificial intelligence

Вступ

Шлях, яким рухається сучасне суспільство, це шлях автоматизації, роботизації, інтелектуалізації процесів, які виконуються людьми, шлях поступової відмови від небезпечних, трудомістких, некомфортних для людини процесів, що вимагають високого рівня концентрації уваги. У рамках руху цим шляхом, уже давно й ґрунтовно, описуються підходи до проєктування всіляких мобільних роботів, для їх ефективного, безперебійного функціонування в різних, часом небезпечних, середовищах [1, 2]. З досвіду моїх колег, що серйозно займаються цим питанням, можна сказати, що розробка й тестування мобільного робота – це трудомісткі й вкрай нетривіальні процеси. Вони вимагають великого обсягу знань у різних сферах, насамперед, у електротехніці та програмуванні, інженерних, конструкторських навичок, вільного орієнтування в технічних новин-

ках, розуміння принципів та технологій, за якими повинен будуватись штучний інтелект, численних експериментів, математичних моделей, проб і помилок. Але щоб побудувати серйозний апарат, особливо такий, що дозволить зберегти життя й здоров'я людей у складних і небезпечних середовищах, одних знань, навичок і прагнення недостатньо. Потрібен такий ресурс як час, багато часу. Потрібні кошти й теж у великих об’ємах. А в цих ресурсах завжди спостерігається дефіцит. Особливо, якщо взяти до уваги той факт, що процес розробки мобільного роботу досить часто суміщено з затратним по часу процесом отримання знань (що не дивно з огляду на те, наскільки розробка мобільних роботів є складним і творчим процесом) чи взагалі є частиною цього процесу, етапом навчання. Кошти – це, на жаль, просто насущна проблема для нашої країни. А без них немає широких можливостей, необхідних деталей, компонентів та конструкцій, немає їх 3D-друку, не кажучи вже про спеціальні випробувальні стенди та втілення «у металі» більших за розміром та потужністю прототипів, ніж найпростіші. Та й у побудові простої, навчальної конструкції мобільного робота доводиться стикатись з такими проблемами, як необхідність використовувати цілу низку складних, не пов’язаних між собою програмних комплексів, численні повторення одних і тих самих дій, випробування правок, внесених у код, безпосередньо на роботі, що створюється (а це знос обладнання та ризик його спалити), нестача компонентів і можливості працювати з ними вільно.

Процеси розробки та тестування мобільних роботів добре піддаються пришвидшенню за рахунок автоматизації, і логічно було б для вирішення описаних проблем сумістити суттєві частини цих розрізнених процесів у одному єдиному програмному комплексі.

Постановка проблеми

Отже, очевидно, що для вирішення задач, які стоять при розробці мобільних роботів, було б доцільно мати єдиний, серйозний програмний комплекс для проєктування та тестування цих роботів (а там, де роботи, там і більш прості роботизовані системи). Особливо, якби цей комплекс реально містив у собі весь необхідний функціонал для процесу розробки мобільного робота й дозволяв взагалі відмовитись або звести до мінімуму використання сторонніх технологій та програм, дозволяв користувачеві створювати необхідні тривимірні й фізичні моделі, проєктувати електричну начинку робота, писати штучний інтелект, настроювати середовище тестування й проводити саме тестування з оперативним внесенням змін у будь-який компонент системи, писати код, який можна буде використовувати на мобільному роботі відразу, без додаткових змін. Правильною була б архітектура, при якій цей комплекс був би максимально легким у використанні та вивченні, розроблявся відразу з прицілом на навчання робототехніці та електроніці, і вза-галі був подібний до гри.

Однак, існуючі рішення досить далекі до цієї ідилії. Програмні комплекси, які присутні на ринку, є складними, вузькоспеціалізованими та слабо підштовхують до творчості. Вони направлені більше на окремі процеси, – такі як тестування штучного інтелекту мобільного робота, відпрацювання окремих алгоритмів. Такі програмні комплекси часто ніяк не поєднані з електричною схемою робота й базуються на готовій, не мають засобів її тестування та моделювання, не мають засобів 3D-моделювання, іноді не мають широких засобів налаштування окремих компонентів, середовищ, відмінних від звичайного наземного, в переважній більшості є складними у навчанні та підготовці спеціалістів і не захищають їх від типових проблем, що неминуче виникають у роботі, є неінтуїтивними, такими, що потребують дуже високого «рівня входу» підготовки користувача, мають неуніверсальний, розподілений функціонал, потребують тісної взаємодії з іншими програмними комплексами. Хоча частина рішень, наприклад, для тестування, моделювання фізичних процесів

та електричних схем може містити досить гарні реалізації.

Імовірно створенню всеосяжного програмного комплексу для проєктування і тестування мобільних роботів перешкоджає велика складність та дорожнеча такої розробки, а може просто немає виразної схеми зведення різних процесів проєктування та тестування мобільних роботів докупи. Можливо розвиток технологій просто випереджає розвиток засобів роботи з ними. Врешті решт ніщо не заважає помріяти на цей рахунок. Можливо, якщо розробити концепцію та описати єдиний програмний комплекс проєктування мобільних роботів, то з'явиться і затребуваність, і економічна доцільність подібної розробки.

Мета дослідження

Провести огляд технологій, що використовуються для процесів проєктування, моделювання і тестування мобільних роботів. Дослідити можливість поєднання їх функціоналу в єдиному програмному комплексі. Розробити концепцію побудови й інтерфейс такого програмного комплексу. Визначити, з чого він повинен складатись та яким повинен бути алгоритм роботи з ним. При цьому, особливий наголос зробити на низький «рівень входу» підготовки користувача й інтуїтивну зрозумілість для новачків, дружність інтерфейсу, за наявності об'єднання основного функціоналу в одному програмному комплексі, на розширені можливості й, зокрема, на моделюван-ня роботи в різноманітних середовищах (повітряне, наземне, підводне).

Огляд існуючих рішень

Перше, з чим стикається розробник мобільного робота, – це необхідність вирішити, які задачі буде виконувати розроблюваний ним апарат і з яких компонентів він буде складатись. Чітке визначення задач може привести до вибору готових рішень, наприклад, для участі в дослідницькій студентській програмі MIT Duckietown [3] знадобиться зовсім простий робот з однією камерою та двома колесами під управлінням Raspberry Pi. Цього робота можна завжди купити у готовому вигляді. А, наприклад, для розробки більш складного 4-х колісного робота-машинки, призначеного для переміщень на природі чи для розробки квадрокоптеру, вже доведеться розбиратись з тим, як скомпонувати окремі мікросхеми, двигуни, датчики, шилди та інше, розбиратись, як усе це правильно вибрати й чому вибір саме такий, і як під'єднати та як не спалити це все відразу ж. З власними рішеннями все складно. Тут лежить перший камінь спотикання, бо для правильного вибору необхідних мікросхем доведеться йти скоріше в глибини інтернету, на форуми, до знаючих людей, у каталоги численних інтернет-магазинів, а не до спеціальних програмних засобів, що могли б дозволити, хоча б і на початковому етапі, створити віртуальну схему робота та погратись з її параметрами, дозволили б легко самостійно визначитись з покупкою необхідних компонентів, мали б суттєві підказки по частині схемотехніки. Звичайно, інтернет, форуми і знаючі люди відразу підкажуть таке чудове сімейство апаратних засобів, призначених для створення простих систем автоматики та робототехніки як Arduino, підкажуть, що якщо потрібен більш серйозний засіб, то слід для початку взяти одноплатний комп'ютер Raspberry Pi, опишуть і інші компоненти робота. Але які є цьому альтернативи? Які є інші плати? Які датчики підібрати? Як розрахувати необхідну потужність джерел живлення? Які обрати двигуни? Як правильно побудувати електричну схему роботу? Взагалі, як працювати з такими засобами як Arduino і Raspberry Pi? Щоб відповісти собі на ці питання нормальним шляхом, є довге і завзяте вивчення різноманітної теорії, характеристик, підходів, маса проб і помилок. І, що найгірше, все це буде в цілій низці різних програмних засобів. Вже на цьому етапі створення досить простого власного робота сучасна робототехніка відштовхує людей. Звичайно, можна використати спеціальні програми-симулятори і, наприклад, для Arduino вже існує досить багато таких засобів. Це спеціалізовані: Virtual BreadBoard, PaulWare Arduino Simulator, Arduino Simulator від Virtronics, Simduino, ArduinoSim, Emulare Arduino Simulator, та такі, які

дозволяють працювати, в тому числі, й з Arduino: EasyEDA, Proteus, AutoCAD 123D, Autodesk Tinkercad, LTspice, Yenka, TINACloud, PSpice, Circuit Lab, Circuits-cloud, Systemvision. Але більшість з них мають або урізані можливості, або дуже «великий поріг входження». Щоб розібратись зі спеціалізованими програмними комплексами, звичайному користувачеві чи студенту потрібно, як мінімум, вивчити схемотехніку до достатньо серйозного рівня й тільки потім переходити до якоїсь творчості. Або працювати безпосередньо з «залізом», експериментувати на реальних платах, датчиках і т. п.

До речі, для того, щоб спроєктувати загальний вигляд робота, що розробляється, частіше за все потрібно спроєктувати платформу, на якій будуть розміщуватись його компоненти, корпус, який їх покриє, потрібно буде змоделювати роботу підвіски, якщо вона там буде, надрукувати це все на 3D-принтері чи зробити вручну. А це необхідність освоювати нові складні технології – технології 3D-моделювання. Для технічно точного моделювання використовуються SolidWorks, AutoCAD, CATIA, Компас. Для моделювання геометрії можуть бути використані й менш націлені на роботу з технічним проєктуванням системи, – 3ds Max, Maya, Cinema 4D, Blender, Sketchup, K-3D, MODO, Art of Illusion і т. п.

Отже, якщо тільки на етапі проєктування робота, вибору його «начинки» і побудови електричної схеми виникає безліч проблем, то проблеми в написанні його штучного інтелекту будуть просто гарантовані. Звичайно, для простих рішень на Arduino всю програмну частину можна помістити в одному скетчі й горя не знати, але для роботи потужного штучного інтелекту все ж потрібен потужний комп'ютер, потрібно писати багато складного коду й бажано, щоб цей код можна було потім використати повторно, потрібне складне тестування. Щоб не ганяти кожний раз створеного робота, тестуючи, наприклад, свою реалізацію такого методу як SLAM, потрібні спеціальні віртуальні симулятори роботів.

Найбільш розповсюдженою готовою операційною системою для роботів, фреймворком, що дає функціональність для розподіленої роботи, є Robot Operating System (ROS). Саме за допомогою цього багатопоточного, кросплатформенного Open Source фреймворка зараз найчастіше вирішується проблема апаратної абстракції, реалізації низькорівневого контролю, повторного використання коду, тощо. До того ж, ROS підтримує найпопулярніші програмні комплекси, що виступають як симулятори, як віртуальні середовища, в яких можна протестувати мобільного робота без необхідності його реалізації «в металі». Вони більш за все наближені до ідеї дослідження, тому роздивимось їх детальніше.

Якщо потрібно щось моделювати, і особливо фізичні процеси, на розум приходить такий пакет для вирішення технічних обчислень як MATLAB (рис. 1) та Simulink [4], який є середовищем графічного програмування і моделювання та базується на основі MATLAB.

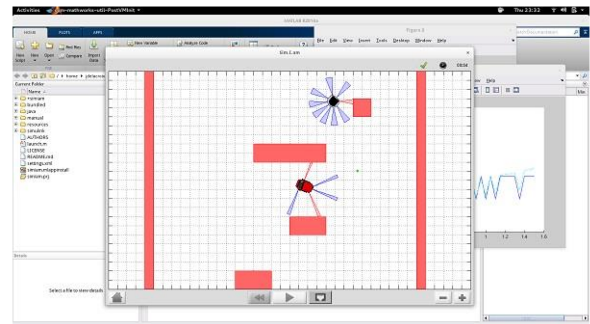


Рис. 1. Приклад проєкту у MATLAB та Simulink

Потужність, точність і математична всеосяжність даних інструментів дозволяє розробляти і тестувати мобільних роботів, але цей процес є досить громіздким, трудомістким та складним у безпосередньому виконанні. Для того, щоб створити і протестувати мобільного робота, потрібно вивчити величезну кількість теорії, досягти високого рівня розуміння роботи даних технологій, плідно попрацювати над налаштуванням об'єктів і сцени. Звичайно MATLAB є незрівнянним у моделюванні фізики, є розповсюдженим, працює з усіма основними операційними системами та підтримує ROS, але він все ж занадто складний і громіздкий, а тому є далеким від ідеалу.

З ROS зазвичай використовується такий симулятор робототехніки як Gazebo [5, 6] (рис. 2). Відкритий код, використання аж чотирьох двигунів для роботи з фізикою (ODE, Bullet, Simbody, DART), здатність до точного моделювання, гнучкість, гарне моделювання роботи різноманітних датчиків роблять цей програмний комплекс популярним. У зв’язці з ROS він навіть використовується у змаганнях з робототехніки Robocup, має велике та активне співтовариство розробників. Раніше Gazebo був частиною середовища ROS, але зараз може використовуватись і як окреме програмне забезпечення. Gazebo повноцінно доступний лише для Linux та підтримує написання коду на C++. За його допомогою можна відносно легко і швидко змоделювати робота, але максимально простого, концептуального.

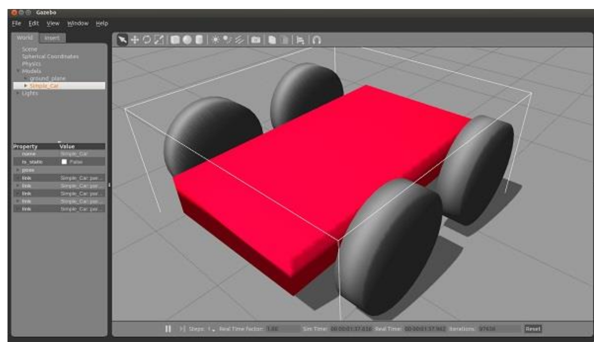


Рис. 2. Приклад проекту у Gazebo

Про якісь рівні роботи з мікросхемами чи моделювання електричного ланцюга робота з готових компонентів тут не йде мова, все ж Gazebo – це середовище для відносно простого віртуального тестування робота.

Також варто звернути увагу на такий розповсюджений програмний комплекс симуляції роботів як Webots [7] (рис. 3). Ця досить давня розробка є кросплатформенною. Вона має дружній графічний інтерфейс, працює з відкритим фізичним двигуном ODE, підтримує написання коду на C, C++, Python, Java, Matlab, має підтримку ROS та може бути пов’язана зі сторонніми додатками за допомогою протоколу TCP/IP. Webots намагається бути повноцінним середовищем моделювання, він використовується як для прикладних, так і для освітніх програм. Наприклад, Webots може працювати разом з OpenCV для ефективної обробки зображень у реальному часі. Але, як і в Gazebo, тут немає мови про роботу з мікросхемами робота, моделюванням його електричного ланцюга, про створення і роботу готових компонентів робота. Все складно і при необхідності моделювати робота у специфічних середовищах, наприклад, під водою, додаток зберігає традиційний для подібних програм високий «поріг входження».

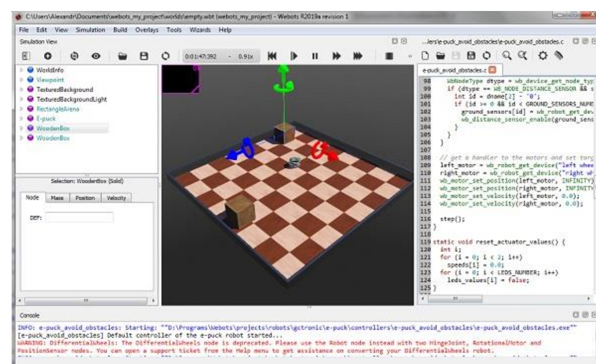


Рис. 3. Приклад проєкта у Webots

Також існує досить вдала платформа розробки експериментальних віртуальних роботів V-REP [8] (рис. 4). Це кросплатформенне програмне забезпечення, яке, як і Webots, претендує на повноцінність у галузі моделювання. V-REP дозволяє індивідуально контролювати кожний об’єкт на сцені за допомогою дочірніх сценаріїв, написання плагінів, вузлів ROS, через роботу з зовнішніми додатками. V-REP працює одразу з чотирма фізичними двигунами (ODE, Bullet, Vortex, Newton), дозволяє зручно працювати з інверсивною кінематикою. Код можна писати на C, C++, Python, Java, Urbi, Matlab, Octave, є підтримка ROS. V-REP має більш розгорнуті можливості з налаштування окремих компонентів, є більш інтуїтивно зрозумілою і легкою в засвоєнні. Однак і недоліки має майже ті самі, що й Webots, – це відсутність можливості працювати з мікросхемами, моделювати електричні ланцюги робота, взагалі його начинку, створювати нові компоненти робота та працювати з їх заготовками. Якщо потрібно працювати у специфічних середовищах з V-REP – це проблема. Хоча, в цілому, вона має більш дружній графічний інтерфейс і краще підходить для задач навчання.

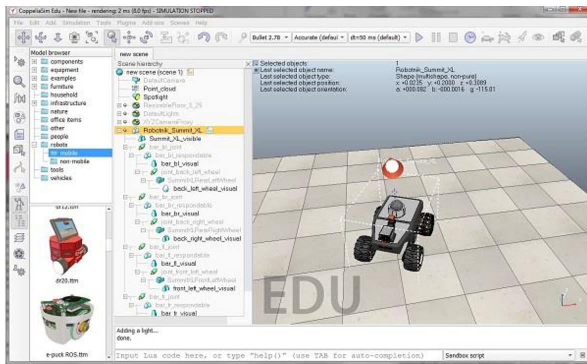


Рис. 4. Приклад проєкта у V-REP

Отже, для того, щоб розробити свого мобільного робота, потрібно гарно освоїти схемотехніку, вирішити для себе низку проблем, які виникають при роботі з нею, конструкторських проблем, мати навички в програмуванні. У таблиці 1 показано порівняння існуючих програмних комплексів, спрямованих на проєктування і тестування мобільних роботів. Якщо проєктування буде проходити віртуально, потрібно вивчити спеціальне, не пов’язане з процесом тестування мобільних роботів, програмне забезпечення зі схемотехніки, яке є досить ізольованим. Якщо потрібно щось більше, ніж тестовий робот з примітивним способом переміщення, потрібно мати знання в проєктуванні, потрібно працювати зі спеціальними програмними комплексами, створеними для цих цілей (SolidWorks та інші). І тут людину тільки починають наздоганяти основні програмні проблеми, такі як: проблема написання адекватного штучного інтелекту, робота з зображенням камери, орієнтування за допомогою цього зображення й інформації з датчиків, тобто інші камені спотикання. Тут постає необхідність використовувати віртуальні симулятори та можливо фреймворк ROS. Останній, завдяки своєму розподіленому підходу, широкому використанню сторонніх засобів симуляції (а в них треба ще розібратись з тим, як використовувати фізику, засвоїти програмування під відповідні фізичні двигуни), проблем тільки додає, та й нормально працює він тільки під Linux, є досить складним у роботі. Пройти всі ці кола пекла і таки створити свого мобільного робота можуть тільки добре вмотивовані люди. Звідси випливає необхідність все ж таки мати універсальний комплекс віртуального моделювання мобільних робототехнічних засобів, який дозволить створювати роботів і без прояву трудового героїзму. Звідси випливають два основних, базових принципи його побудови: максимальна ігрова простота використання та освоєння і необхідність будувати процес розробки так, щоб він проходив виключно від легкого до складного, бажано шляхом поступового ускладнення вже промодельованих процесів і систем.

Концепція

Отже, жоден програмний комплекс не містить у собі повного переліку засобів (таблиця 1, ліва колонка) для того, щоб можна було спроєктувати мобільного робота, відтестувати в віртуальному середовищі роботу його електросхем, виявити в них можливі помилки, не змінюючи програмного засобу, написати штучний інтелект для робота та провести його тестування в віртуальному середовищі. Немає таких програмних комплексів, які б робили ставку на експериментування, на інтуїтивне вивчення, дозволяли створити необхідних роботів та їх компоненти парою кліків миші, мали гнучку систему поглиблення спершу примітивної реалізації, виходу її на режим серйозної симуляції, поєднували це з існуючими можливостями з тестування роботи спеціальних алгоритмів. Навпаки, чим сильніша спеціалізація програмного комплексу і близькість до задач моделювання роботів, тим менш інтуїтивними й зрозумілими виявляються його інтерфейс та спосіб застосування. Тому дане дослідження й вивчає можливість побудови віртуальної лабораторії, що має бодай мінімальну кількість засобів для моделювання електричних схем, для моделювання й, що більш важливо, конструювання власних моделей складних електричних компонентів та плат.

Таблиця 1. Порівняння існуючих програмних комплексів спрямованих на проєктування й тестування мобільних роботів

Назви програм, що проаналізовано *Параметри порівняння*	*MATLAB та Simulink*	*Gazebo*	*Webots*	*V-REP*
Можливість писати код безпосередньо та наявність віртуального майданчика для тестування	+	+	+	+
Можливість легко використати написаний код на розповсюджених платах	+	+	+	+
Просунуте моделювання необхідних фізичних процесів та взаємодії між об'єктами	+	+	+	+
Можливість вибору середовища (гідро, атмосфера, наземне, підземне, ближній космос)	-	-	-	-
Дружній інтерфейс	+	±	+	+
Легкість засвоєння та орієнтація на навчання	-	-	-	±
Спрощений «логічний» режим для навчання і швидкого налаштування компонентів сцени	-	-	-	-
Рівень моделювання електроніки та мікросхемотехніки	+	-	-	-
3D-моделювання компонентів мобільного робототехнічного комплексу (РТК) та елементів оточення	-	±	-	±
Компонування РТК з готових компонентів та їх докладне налаштування	±	-	-	-

Так яким повинен бути програмний комплекс, що буде включати в себе такі можливості? Як можна розширити базові принципи його побудови? З огляду на підхід, який був обраний вище, можна дійти висновку, що програмний комплекс для моделювання мобільних роботів обов'язково повинен бути:

- простим в освоєнні і роботі;
- але при цьому глибоким, достатнім для виконання типових задач;
- побудованим за модульним принципом;
- кросплатформенним;
- таким, що розвивається, використовуючи ідеї своїх користувачів, та проєкти, що вони створюють, дає доступ до цих проєктів;
- цікавим;
- інтуїтивно зрозумілим;
- максимально універсальним;
- мотивуючим до подальшого розвитку одержуваних результатів;
- з широкими можливостями по моделюванню робота;

- робота повинна відповідати руху від простого до складного;
- інтерфейс і вся логіка програмного комплексу повинна підштовхувати користувача до експериментування і наукового пошуку.

Для цього пропонується розділити побудову мобільних роботів на два принципових рівні: простий логічний (рівень, що реалізує нескладний, приблизний, наближений до гри, доступний за замовчуванням функціонал) та складний фізичний (рівень, що містить більш точну реалізацію необхідного функціоналу). За допомогою такого підходу можна отримати результат, подібний до того, що дають комп'ютерні ігри, коли людині цікаво, коли прокидається творчість та бажання ускладнювати отриману реалізацію, коли можна одночасно вчитися й виконувати необхідну роботу. Це логічно, адже можна пропустити складне, можна сконцентруватися спершу на тому, що більш цікаве, що відповідає моменту. Якщо ж при цьому програмний комплекс бодай частково буде володіти такою важливою характеристикою як всеосяжність, якщо за його допомогою можна буде спершу розробити концепцію майбутнього мобільного робота, можна буде забезпечити її прямо зі старту штучним інтелектом (який можна задати готовими шаблонами чи програмним кодом, який, якщо він вже буде написаний, може бути легко помістити у реального робота), якщо можна буде, не змінюючи нічого, перейти до конструювання електричної схеми мобільного робота, до експериментального підбору його характеристик, наприклад, потужності двигуна, ємності джерел живлення, до корегування його ваги, конструкційної міцності тощо, то у підсумку можна значно спростити процес розробки, уникнути багатьох типових помилок. Розбиття процесу моделювання на два рівні також дозволить одразу перейти до тестування, одразу задіяти творчість людини і тільки потім поступово показати проблеми реальної, фізичної реалізації. А, виконуючи цю реалізацію, зостатись у одному єдиному програмному комплексі, чим спростити процес розробки і загальне враження від процесу.

Розроблюваний програмний комплекс (приклад інтерфейсу на рис. 5) повинен складатись з наступних рівнів:

- Рівень тестування програмно і безпосередньо у тривимірному середовищі.
- Рівень моделювання тривимірних об'єктів (у тому числі під майбутній друк на 3D-принтері) і їх редагування.
- Рівень конструювання та налаштування своїх логічних компонентів.
- Рівень створення своїх тривимірних сцен і докладного налаштування тестування.
- Рівень конструювання конкретних мікросхем.
- Рівень управління проєктом.
- Рівень написання програм.
- Рівень роботи зі штучним інтелектом і нейронними мережами (конструктори нейронних мереж, нехай і у мінімальному вигляді, з оглядкою на проблеми навчання і тестування [9]).
- Рівень оформлення документації та ідей.
- Рівень роботи з фізикою і матеріалами (налаштування параметрів тестування, взаємодії об'єктів і фізики процесів, налаштування матеріалів для створення роботів).
- Рівень моделювання роботи механізмів.
- Рівень моделювання роботи з інверсивною кінематикою.

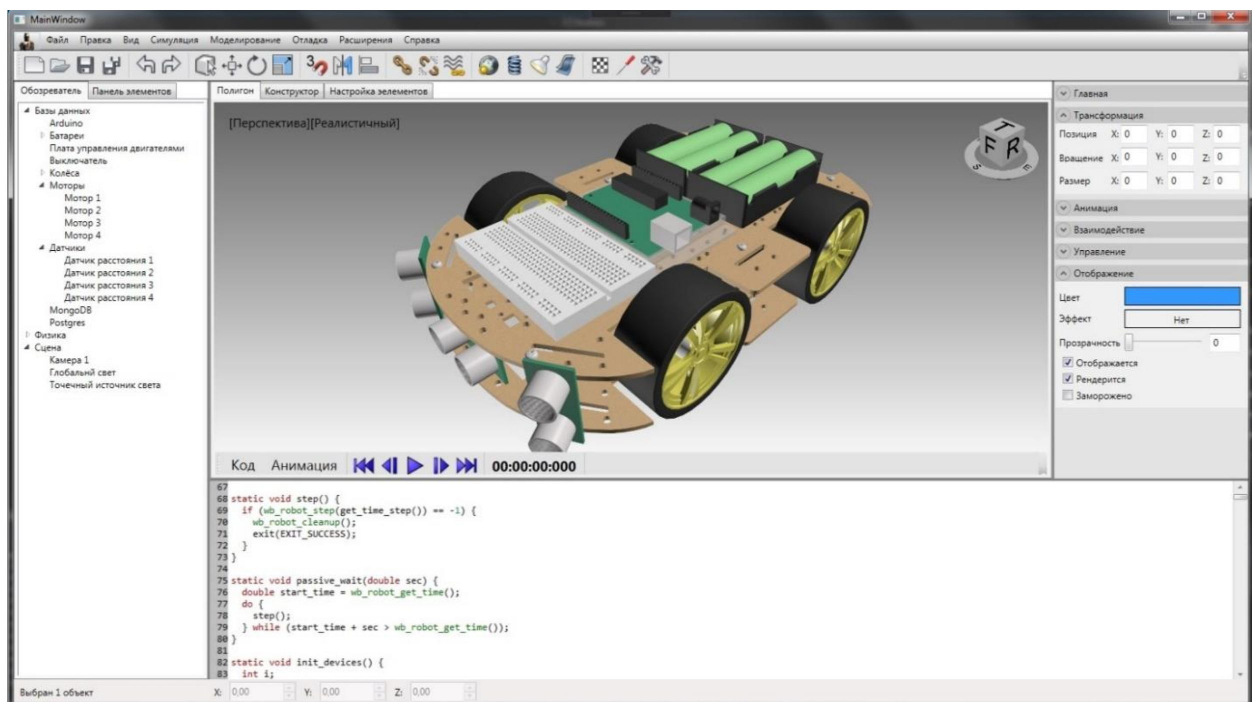


Рис. 5. Перше наближення інтерфейсу універсальної системи комп'ютерного віртуального моделювання мобільних робототехнічних засобів

Висновки

Численні програмні засоби, що використовуються для проєктування, моделювання і тестування мобільних роботів, не є універсальними і максимально всеосяжними. Крім позитивної складової від наявності можливостей, що вони забезпечують, вони володіють також і великим негативним потенціалом для того, щоб складністю та дешевістю своїх рішень відштовхнути дослідників від творчого процесу розробки власних мобільних роботів і робототехнічних засобів, зробити процес розробки дуже довгим і складним.

У той же час ідея створення універсального програмного комплексу, що об'єднав би у собі можливості описаних програмних засобів, видається можливою. У статті запропоновано концепцію побудови такого програмного комплексу, описано, яким він повинен бути, з яких логічних рівнів він повинен складатись та наведено скриншот першого наближення його інтерфейсу.

Література

1. Писаренко Ю.В., «Віртуальне проектування інтелектуалізованих роботів для розвідки і нейтралізації небезпечних екологічних подій»: автореф. дис... канд. техн. наук: 05.13.06, НАН України. Ін-т кібернетики ім. В.М.Глушкова, Київ., 20 с, 2006.
2. Мелкумян К.Ю., «Автоматизована система моніторингу типових техно-екологічних подій»: автореф. дис... канд. техн. наук: 05.13.07, НТУУ «Київ. політехн. ін-т», Київ., 20 с, 2011.
3. Сайт дослідницької програми Duckietown від MIT: https://www.duckietown.org/
4. Документація по MATLAB і Simulink: https://docs.exponenta.ru/
5. Nathan Koenig, Andrew Howard. Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator - Intelligent Robots and Systems, 2004.
6. Документація по Gazebo: http://gazebosim.org/tutorials
7. Olivier Michel, Cyberbotics Ltd. WebotsTM: Professional Mobile Robot Simulation, International Journal ofAdvanced Robotic Systems, 2004.
8. Eric Rohmer, Surya P. N. Singh, Marc Freese, V-REP: A versatile and scalable robot simulation framework, International Conference on Intelligent Robots and Systems, Tokyo, Japan, 2013.
9. Хайкин С., «Нейронные сети: полный курс», Второе издание. Перевод с английского, Изд. дом «Вильямс», Москва – С. - Пб. – Киев, 1104 с., 2006. URl: http://diggerdnepr.ddns.net/wp-

content/uploads/2019/05/khaykin_s_neyronnye_seti_polny_kurs_izd_2_2006_ru_pdf.pdf

References

1. Pisarenko J.V., «Virtual designing intellectualized robots for investigation and neutralization of dangerous ecological incidents»: Abstract of dissertation for Ph.D. technical sciences degree by specialty 05.13.06, V.M. Glushkov Institute of Cybernetica of The National Academy of Sciences of Ukraine, Kiev, 20 p., 2006.
2. Melkumian E.U., «Automated system for monitoring typical techno-ecological accidents»: Abstract of dissertation for Ph.D. technical sciences degree by specialty 05.13.07, National Technical University of Ukraine «Kyiv Polytechnic Institute», Kiev, 20 p., 2011.
3. The MIT Duckietown Research Program Website: https://www.duckietown.org/
4. Documentation on MATLAB and Simulink: https://docs.exponenta.ru/
5. Nathan Koenig, Andrew Howard. Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator - Intelligent Robots and Systems, 2004.
6. Gazebo Documentation: http://gazebosim.org/tutorials
7. Olivier Michel, Cyberbotics Ltd. WebotsTM: Professional Mobile Robot Simulation, International Journal ofAdvanced Robotic Systems, 2004.
8. Eric Rohmer, Surya P. N. Singh, Marc Freese, V-REP: A versatile and scalable robot simulation framework, International Conference on Intelligent Robots and Systems, Tokyo, Japan 2013.
9. Haykin S., «Neural Networks: A Comprehencive Foundation», Second Edition, «Prentice Hall, Inc.», Upper Saddle River, New Jersey 07458, 1104 p., 1999 [On-line]. Available: https://ru.b-ok2.org/book/2436979/070dce

RESUME

O.S. Koval

Concept of system of computer virtual simulation of mobile robototechnical means of technical and environmental events neutralization and solving problems of professional training

The process of virtual computer simulation of mobile robotic systems in its present form is complex, lengthy, requiring a large amount of diverse knowledge and skills, and therefore may be repulsive to potential developers of such systems.

The article provides an overview of the technologies used for the processes of designing, modeling and testing of mobile robots, shows the niche of these software products. For example, the technology of working with electrical circuits in general (EasyEDA, Proteus, AutoCAD 123D, Autodesk Tinkercad, LTspice, LTspice, Yenka, TINACloud, PSpice, etc.), technologies designed to work with special boards, such as Arduino (Virtual BreadBoard, PaulWare Arduino Simulator, Arduino Simulator from Virtronics, Simduino, ArduinoSim, etc.), 3D-design technologies for building the necessary enclosures and mechanics (SolidWorks, AutoCAD, CATIA, Compass), their analogues are less focused on working with technical design (3ds Max, Maya, Cinema 4D, Blender, Sketchup, etc.), is an operating system for the development of ROS robots, designed to simplify the development of such devices, it is a technology of simulating the operation of a particular apparatus, testing his artificial intelligence in virtual three-dimensional environment (MATLAB and Simulink, Gazebo, Webots, V-REP and others).

On this basis, this article explores the idea of creating a universal software package that combines the capabilities of the described software. In particular, the concept of constructing such a software complex is described, describing what it should be ideological, what logical levels it should consist of. It is suggested in the construction of such a software complex to use approaches that have proven themselves in computer games, namely those that simplify the work process as much as possible and, importantly, the process of learning, by dividing the tasks into two levels: logical (maximally simplified, conceptual the level of development at which it goes quickly and according to ready-made templates) and the physical (which has a detailed and accurate implementation, which can be obtained by extending the functionality already created at the logical level). In addition, the first, working approximation of the future interface of this described software package for virtual modeling of mobile robots was given.

Надійшла до редакції19.09.2019